

Új elvű számítási modellek

- Korábban láttuk: a **nemdeterminisztikus Turing-gép** tekinthető olyan eszköznek, ami képes egy bemeneten az összes lehetséges számítás párhuzamos végrehajtására
- A természetben is megfigyelhető **nagyfokú párhuzamosság**: az élő sejtek nagyon nagy számú alkotóelemei egyszerre változhatnak adott körülmények között
- A fenti megfigyelésen alapulnak az olyan **biomolekuláris számítások**, mint például a
 - Membrán számítások
 - DNS számítások vagy a

L-rendszerek

- **L-rendszerek:** A természetben megfigyelhető folyamatok formális nyelvi modellezésének egyik legkorábbi példája
 - Aristid Lindenmayer, 1968
- Hogyan működnek?
 - Egyszerű **szabályok iterált** alkalmazásával modellezik azt, ahogy bizonyos organizmusok (leginkább fák, növények) a növekedés során komplex alakzatokat hoznak létre

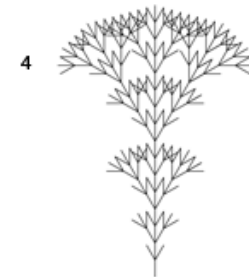
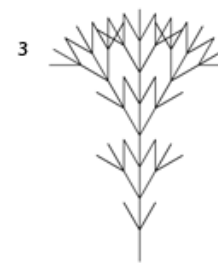
- Példák L-rendszerekkel generált képekre (forrás: Wikipedia):



L-rendszerek

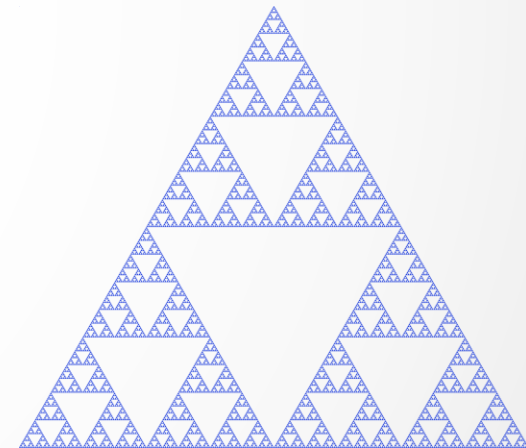
Mit jelent az iterálás?

- Egy bizonyos **mintát ismétlünk** kapott alakzat élein, minden élen egyszerre:



Az iterációt tetszőleges határig növelve egy **fraktált** kapunk

- A fraktálok olyan végtelenül komplex matematikai alakzatok,
 - melyekben egy kisebb részt felnagyítva ugyanolyan struktúrát kapunk, mint a teljes rész és
 - melyekben legalább egy matematikai módon leírható ismétlődés található
 - Egy ismert fraktál a **Sierpinski-háromszög**:



L-rendszerek – Formális definíció

- Egy **L-rendszer** egy olyan $L = (V, S, R)$ párhuzamos átíró rendszer, ahol
 - V : a rendszer **szimbólumainak** ábécéje
 - $S \in V^+$: **axióma**
 - $R: a \rightarrow u$ alakú **szabályok** ($a \in V, u \in V^*$) véges halmaza
 - (feltesszük, hogy minden a szimbólumra pontosan egy a baloldalú szabály van, ha más nem akkor az $a \rightarrow a$)
- Egy L-rendszer egy **iterációs lépése**: átírjuk az aktuális mondatforma **minden betűjét** a neki megfelelő szabály jobb oldalára
- **Példa**: legyen $L = (V, S, R)$ ahol
 - $V = \{A, B\}, S = A, R = \{A \rightarrow B, B \rightarrow BA\}$
 - Az n -ik ($n = 0, 1, 2, \dots$) lépésben a következő mondatformákat kapjuk:
 - A, B, BA, BAB, BABBA, BABBABAB,...
 - Ezek a **Fibonacci szavak**: $S_0 = A, S_1 = B$ és minden $n \geq 2$ -re, $S_n = S_{n-1}S_{n-2}$

L-rendszerek – Fraktálok generálása

- Tegyük fel, hogy képesek vagyunk irányítani egy **rajzoló-eszközt**:
 - rajzolj egy egységni vonalat
 - fordulj el jobbra/balra d fokot
- Ekkor megfelelő **utasítások sorozatával** komplex ábrákat tudunk rajzoltatni ezzel az eszközzel
- Tekintsük a következő szimbólumokat és a nekik megfelelő utasításokat
 - F, G : rajzolj egy egységet
 - $+$ fordult jobbra
 - $-$ fordulj balra (egy előre beállított szöget)
- **Példa**: legyen a beállított szög 45 fok
 - akkor az $F - F + F$ -nek megfelelő ábra:



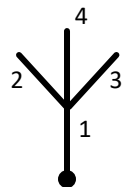
L-rendszerek – Fraktálok generálása

- Vonal **elágaztatása**:

- [- elmenti egy verem tetejére az írófej aktuális pozícióját és irányát
-] - leveszi a verem tetejéről az írófej pozícióját és irányát, és beállítja azt

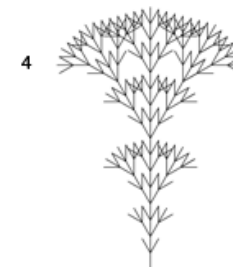
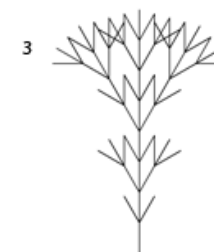
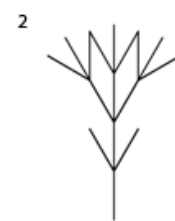
- **Példa**:

- Az $F[-F][+F]F$ -nek megfelelő ábra (a vonalak rajzolási sorrendjét is feltüntettük):



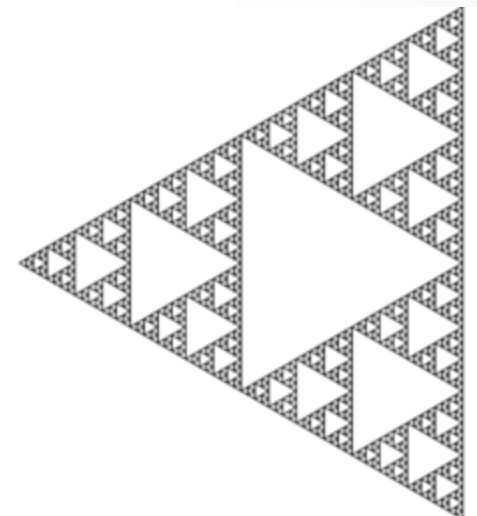
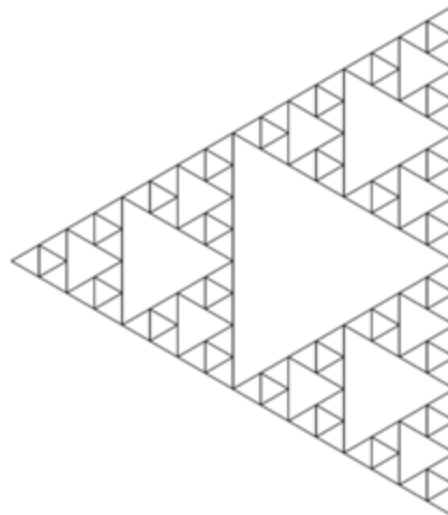
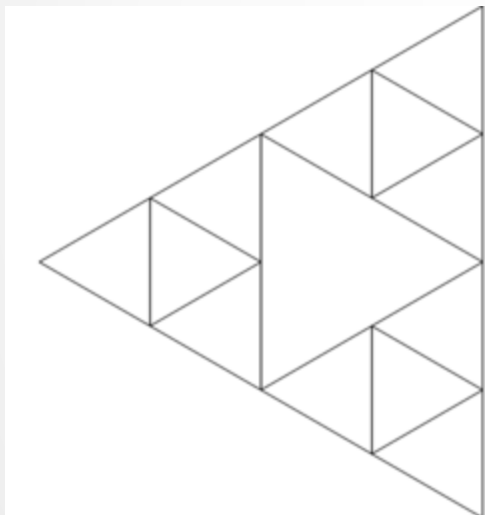
- **Példa**: vegyük a következő L-rendszert

- $L = (V, S, R)$ ahol
- $V = \{F, +, -, [,]\}$, $S=F$,
- $R = \{F \rightarrow F[-F][+F]F\} \cup \{a \rightarrow a \mid a \in \{+, -, [,]\}$
- A rendszer $i = 1, 2, 3, 4$ iterációja után a kapott mondatformák **22.5 fokos** szöggel kirajzoltatva:



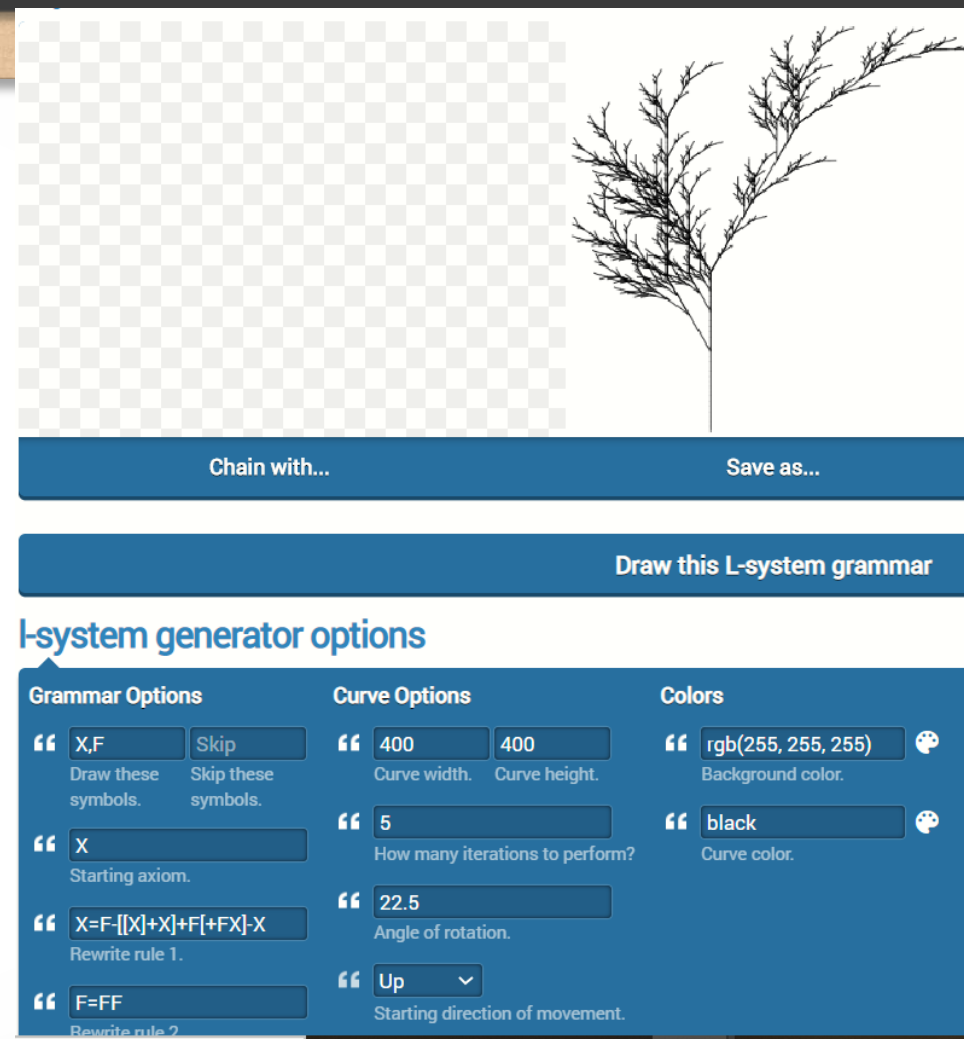
L-rendszerek – Fraktálok generálása

- **Példa:** vegyük a következő L-rendszert
 - $L = (V, S, R)$ ahol
 - $V = \{F, G, +, -\}$, $S = F - G - G$, $R = \{F \rightarrow F - G + F + G - F, G \rightarrow GG, + \rightarrow +, - \rightarrow -\}$
 - A rendszer $i = 2, 3, 4$ iterációja után a kapott mondatformák 120 fokos szöggel kirajzoltatva (a Sierpinski-háromszög):



L-rendszerek – Online szimulátor

- Webes eszköz:
<https://onlinemathtools.com/l-system-generator>
- Érdemes kipróbálni a következőket is:
 1. szög: 22.5, axióma: F,
szabály: $F=FF-[-F+F+F]+[+F-F-F]$
 2. szög: 25.7, axióma: F
szabály: $F=F[+F]F[-F]F$
 3. szög: 20, axióma: VZFFF
szabályok:
 $V=[+++W][---W]YV$
 $W=+X[-W]Z$
 $X=-W[+X]Z$
 $Y=YZ$
 $Z=[-FFF][+FFF]F$
iteráció: 9



EOL-rendszerek

Az **EOL-rendszerek** az L-rendszerek általánosításai

Egy EOL-rendszer egy olyan $G = (V, \Sigma, S, R)$ párhuzamos átíró rendszer, ahol

- V : a rendszer **szimbólumainak** ábécéje
- $\Sigma \subseteq V$ a **terminálisok** ábécéje
- $S \in V^+$: **axióma**
- $R: a \rightarrow u$ alakú **szabályok** ($a \in V, u \in V^*$) véges halmaza
 - (feltesszük, hogy minden a szimbólumra **legalább** egy a baloldalú szabály van)

Egy L-rendszer egy **iterációs lépése**: átírjuk az aktuális mondatforma **minden betűjét** a neki megfelelő szabály jobb oldalára; jele: \Rightarrow

Egy G L-rendszer által **generált nyelv**: azon csak terminálisokból álló szavak halmaza, melyek iterációs lépések sorozatával megkaphatók S -ből

- A G által generált nyelvet $L(G)$ -vel jelöljük

EOL-rendszerek – Példa

Legyen $G = (V, \Sigma, S, R)$ a következő EOL-rendszer:

- $V = \{A, A', a, B, B', b, C, C', c, F\}, \Sigma = \{a, b, c\}$
- $S = ABC$
- $R = \{ A \rightarrow AA', A \rightarrow a, A' \rightarrow A', A' \rightarrow a, a \rightarrow F$
 $B \rightarrow BB', B \rightarrow b, B' \rightarrow B', B' \rightarrow b, b \rightarrow F$
 $C \rightarrow CC', C \rightarrow c, C' \rightarrow C', C' \rightarrow c, c \rightarrow F$
 $F \rightarrow F \}$
- Példák iterációkra:
 - $ABC \Rightarrow AA'BB'CC' \Rightarrow AA'A'BB'B'CC'C' \Rightarrow aaabbbccc$
 - $ABC \Rightarrow AA'BB'CC' \Rightarrow aaBB'B'CC'C' \Rightarrow FFbbbccc$
- A generált nyelv: $L(G) = \{a^n b^n c^n \mid n \geq 1\}$

EOL-rendszerek – Számítási erő

Állítás: Minden környezetfüggetlen (CF) nyelv generálható egy EOL-rendszerrel

- Legyen L egy CF nyelv és $G = (N, \Sigma, P, S)$ egy L – et generáló környezetfüggetlen nyelvtan
- Legyen $G' = (V, \Sigma, A, R)$ a következő EOL-rendszer:
 - $V = N \cup \Sigma$, $A = S$, $R = P \cup \{a \rightarrow a \mid a \in \Sigma\}$
- Belátható, hogy a $L(G') = L(G)$

Állítás: Van olyan nyelv, ami generálható EOL-rendszerrel, de nem környezetfüggetlen

- Az előző fólián látott $\{a^n b^n c^n \mid n \geq 1\}$ nyelv nem környezetfüggetlen

Következmény: $CF \subset EOL$, ahol EOL az EOL-rendszerekkel generálható nyelvek osztálya

Megjegyzés: $EOL \subset CS$

Membrán számítás

- A membrán számítást **Gheorghe Păun** vezette be 1998-ban
- Azóta egy széles körben kutatott, gyorsan fejlődő tudományággá vált
- A biomolekuláris számítások azon ága, ahol az élő sejtekben végbemenő folyamatokon alapulva adunk meg és vizsgálunk új, absztrakt számítási modelleket
- A vizsgált modelleket **membrán rendszereknek** (másnéven **P rendszereknek**) nevezzük
- A membrán rendszerek tehát a biológiai sejtek absztrakt modelljei
- A membrán rendszerek számos modellváltozata ismert (**transition**, symport-antiport, spiking neural, **aktív membrános**)

Membrán rendszerek

Biológiai
sejt



absztrakció

- Egy membrán rendszer membránok által határolt régiók hierarchikus rendszere
- A régiókban objektumok multihalmazai vannak
- A rendszer működése a régiókhoz rendelt szabályok alapján történik
- Ezek a szabályok objektumokat írhatnak át vagy küldhetnek a membránokon keresztül
- A szabályok alkalmazása nemdeterminisztikusan és maximálisan párhuzamos módon történik

Membrán
rendszer

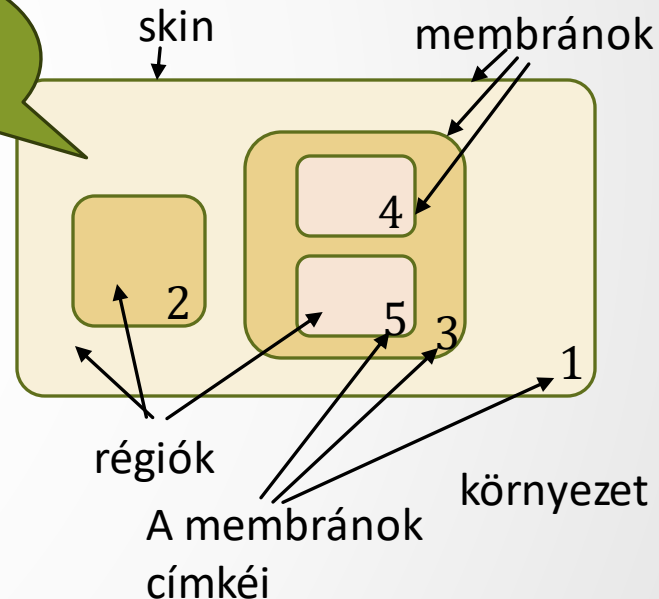
skin

membránok

régiók

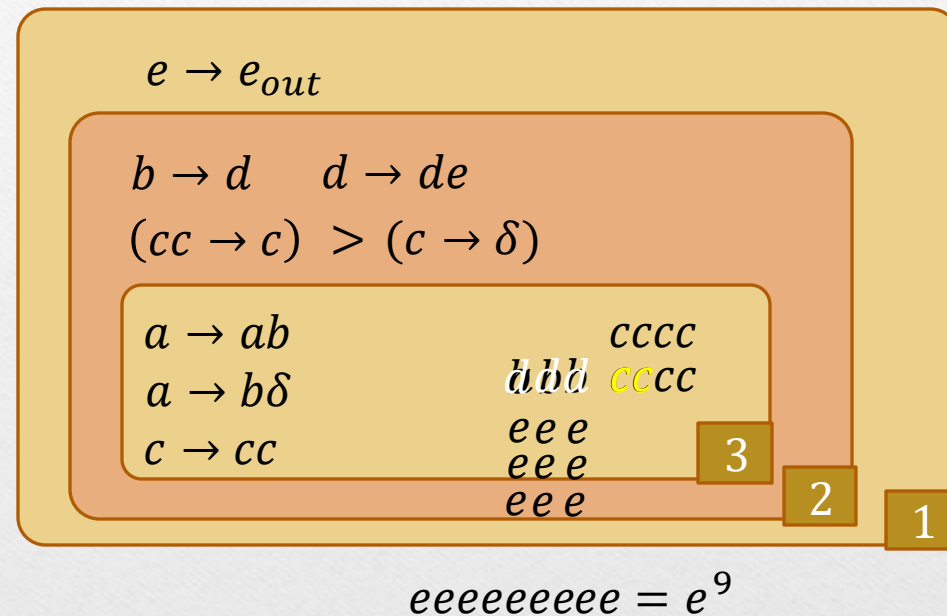
A membránok
címkéi

környezet



Membrán rendszerek – Bevezető példa

Egy klasszikus transition membrán rendszer működése:



Ez a rendszer az e, e^4, e^9, \dots sorozatot számolja ki